United States Patent Application

In the name of: Stefan Pleisch and Andre Schiper

Entitled: A FAULT-TOLERANT MOBILE AGENT FOR A COMPUTER NETWORK

Filed: March 29, 2001

IBM Docket No.: CH9-2000-0077

Express Mail Label: EL 847582946 US
Date of Deposit: 3/29/01

A Fault-Tolerant Mobile Agent for a Computer Network

Field and Background of the Invention

The invention relates to a method of operating a mobile agent that travels through a network of a number of computers.

Such a mobile agent system is known, e.g. from A. Mohindra, A. Purakayastha and P. Thati: Exploiting non-determinism for reliability of mobile agent systems", in Proc. of the Int. Conf. On Dependable Systems and Networks, pages 144 – 153, New York, June 2000.

One concern in connection with such a mobile agent system is the fact that failures may lead to blocking or a complete loss of the mobile agent. This problem may be solved by replication of the mobile agent. However, this leads to the so-called exactly-once execution problem which has to be fulfilled. In the above mentioned prior art document, this problem is solved by detecting multiple mobile agents at the end of any execution and by undoing all effects of multiple executions. However, such an undoing function is not simple and often limits the overall system throughput.

Summary of the Invention

It is an object of the invention to provide a method of operating a mobile agent which is fault-tolerant without being too complex.

This object is solved by one aspect of the present invention, which provides a method of operating a mobile agent that travels through a network of a number of computers, wherein the mobile agent is executed in a sequence of stages and wherein each stage comprises a set of places, the method comprising the following steps: executing the mobile agent in at least one of the set of places of a respective one of the stages, evaluating in which place of the respective stage the mobile agent has been executed successfully, agreeing on this place among the set of places, aborting and/or undoing any operation in connection with the mobile agent in any other place of the respective stage, and moving the modified mobile agent resulting from the successful

execution to the next stage.

As well, this object is solved by the computer program product that contains instructions implementing the steps of the foregoing method, and still further, whereby the foregoing method steps are managed by a fault-tolerance enabler (FTE) which is independent of the mobile agent.

The invention uses the replication of the mobile agent so that a set of places is available within a sequence of stages in which the mobile agent is executed. In order to prevent blocking and to solve the exactly-once execution problem, the invention includes the idea to model the execution of the mobile agent and its replication as a sequence of agreement problems.

According to the invention, the mobile agent is executed in at least one of the set of places of a respective one of the stages. Then, it is evaluated in which place of the respective stage the mobile agent has been executed successfully. After this step, any operation in connection with the mobile agent in any other place of the respective stage is aborted and/or undone. Finally, the modified mobile agent resulting from the successful execution is moved to the next stage.

This method ensures that only exactly one execution of the mobile agent within the set of places of the respective stage is committed whereas all other possible executions are aborted and/or undone.

The implementation of the inventive method may preferably be done by a so-called fault-tolerance enabler (FTE) which may be programmed as an independent component but which may then travel to the places of the stages together with the mobile agent.

Further advantages and embodiments of the invention are apparent from the further claims and/or from the following description of the drawings.

Brief Description of the Drawings

Examples of the invention are depicted in the drawings and are described in detail below by way

of example. It is shown in

Figure 1a: a schematic representation of a method of operating a mobile agent according to an embodiment of the invention;

Figure 1b: a schematic representation of the method of figure 1a comprising a failure;

Figure 2: a schematic block diagram of a consensus method according to an embodiment of the invention; and

Figure 3: a schematic block diagram of an architecture of the mobile agent according to an embodiment of the invention.

All the figures are for sake of clarity not shown in real dimensions, nor are the relations between the dimensions shown in a realistic scale.

Detailed Description of Embodiments of the Invention

In the following, the various exemplary embodiments of the invention are described.

A mobile agent is a computer program that acts autonomously on behalf of an agent owner or user and that travels through a network of a number of computers. Failures in such a system may lead to a blocking of the execution of the mobile agent or to a partial or complete loss of the mobile agent. As well, the agent owner often does not know whether the mobile agent is actually lost due to the failure or whether its execution has only been delayed due to slow computers. The agent owner may then believe that the mobile agent has been lost when in fact it has not been, or he waits for the mobile agent to finish when it has failed.

This uncertainty may be removed by a mobile agent with a fault-tolerant execution. The mobile agent then either reaches its destination or at least notifies a problem.

Such fault-tolerance may be gained by replicating the mobile agent. Replication of the mobile agent is similar to the addition of redundancy and enables the mobile agent to continue its execution despite failures. The blocking of the mobile agent, therefore, is prevented.

However, the replication of the mobile agent may lead to the violation of the so-called exactly-once execution property of the execution of the mobile agent. If, for example, a mobile agent is executed on a first computer and fails, then the first computer may survive, however, comprising modifications performed by the failing mobile agent. A replication of the mobile agent is then executed on a second computer performing modifications of the second computer. This results in modifications in the first and the second computer which contradicts the exactly-once execution property. This property is also violated if the failure of a mobile agent is detected, however, the mobile agent has actually not failed. In this case, the unreliable failure detection leads to a double execution of the mobile agent which, as mentioned, contradicts the exactly-once execution property.

The idea is to model the execution of the mobile agent and its replication as a sequence of agreement problems. For that purpose, the following assumptions are taken and explained now in connection with figure 1a.

As already described, a mobile agent a_i executes on a sequence of computers; wherein $i = 0 \ldots n$. A place p_i provides a logical execution environment for the mobile agent a_i wherein each computer may host multiple places p_i. The execution of the mobile agent a_i at a place p_i is called a stage S_i. The replicas of the mobile agent a_i execute on different places p_i^j within one and the same stage S_i. Two stages S_i and S_{i+1} are separated by a move operation of the mobile agent a_i. The places p_i^j where the first and the last execution of the mobile agent a_i take place are called the source p_0^0 and the destination p_n^0 of the mobile agent a_i which may be identical.

According to figure 1a, the mobile agent a_0 is executed in the place p_0^0 of stage S_0 which is the source of the mobile agent. Then, after successfully executing the mobile agent a_0, the agreement problem is solved by a decision $<a_1, M_1>p_0^0$ in which a_1 is the resulting mobile agent after executing the mobile agent a_0 at the place p_0^0 of the stage S_0, M_1 is the set of places p_1^j of the next

stage S_1, and p_0^0 is that place of the stage S_0 which has successfully executed the mobile agent a_0. The evaluation of the aforementioned decision will be explained later.

Due to this decision, the mobile agent a_1 enters the next stage S_1 at the place p_1^0 and is executed there. According to figure 1a, the stage S_1 comprises the further places p_1^1, p_1^2 and p_1^3 in which replicas of the mobile agent a_1 may be executed. However, after successfully executing the mobile agent a_1 at place p_1^0 of the stage S_1, the agreement problem is solved at once, i.e. it is agreed among the set M_1 of places p_1^0, p_1^1, p_1^2 and p_1^3 that the place p_1^0 has executed the mobile agent a_1 successfully. This leads to a decision $<a_2, M_2>p_1^0$ in which a_2 is the resulting mobile agent after executing the mobile agent a_1 at stage S_1, M_2 is the set of places p_2^j of the next stage S_2, and p_1^0 is that place of the stage S_1 which has successfully executed the mobile agent a_1.

According to figure 1a, this procedure is continued through the sequence of stages S_i until the destination of the mobile agent is reached. There, the mobile agent a_4 enters the stage S_4 and is executed in the only place p_4^0.

In figure 1a, no failure occurs. This means that none of the computers fails, none of the places fails, and the execution of none of the mobile agents fails. Moreover, no incorrect failure detection is present. Therefore, the mobile agent is always executed in the first place of any of those stages which comprise more than one place, i.e. in the places p_1^0, p_2^0 and p_3^0 of the stages S_1, S_2 and S_3. Therefore, these places p_1^0, p_2^0 and p_3^0 are also part of the respective decision after the execution of the mobile agents in the respective stages.

In contrast thereto, figure 1b comprises a failure of the place p_2^0 of the stage S_2. This is depicted in figure 1b with the expression "crash".

When the place p_2^1 detects the failure of the place p_2^0, it executes a replica of the mobile agent a_2. It has to be mentioned that the place p_2^0 is the first one in the sequence of the set M_2 of the places p_2^0, p_2^1, p_2^2 and p_2^3 of the stage S_2 which executes the mobile agent a_2. The next place p_2^1 is able to monitor the execution of the mobile agent a_2 in the preceding place p_2^0. Upon detection of a failure of the mobile agent a_2 or the place p_2^0, the next place p_2^1 starts executing the replica of the

mobile agent a_2.

After successfully executing the replica of the mobile agent a_2 in the place p_2^1 of the stage S_2, the agreement problem is solved. It is agreed among the set M_2 of places p_2^0, p_2^1, p_2^2 and p_2^3 in which place the mobile agent has been executed successfully. As described, this is the place p_2^1. This leads to a decision $<a_3, M_3>p_2^1$ in which a_3 is the resulting mobile agent after executing the mobile agent a_2 at stage S_2, M_3 is the set of places p_3^j of the next stage S_3, and p_2^1 is that place of the stage S_2 which has successfully executed the mobile agent a_2.

The important difference between figure 1a and figure 1b, therefore, is that the decision after stage S_2 of figure 1b comprises the place p_2^1 as successfully executing the mobile agent a_2 whereas the decision after the stage S_2 of figure 1a comprises the place p_2^0. The decision of figure 1b, therefore, recognizes the fact that the execution of the mobile agent a_2 failed in the place p_2^0 of stage S_2 of figure 1b.

The decisions that are taken in each of the stages S_i of the figures 1a and 1b are evaluated by using a consensus method which will be explained now in connection with figure 2.

Figure 2 shows a stage S_i which may be any of the stages shown in figures 1a and 1b. The stage S_i comprises the corresponding mobile agent a_i and a so-called fault-tolerance enabler (FTE) as two independent components.

If the stage S_i is entered from a preceding stage, the FTE starts to solve the agreement problem for this stage S_i (see block 20). For that purpose, the block 20 initiates (see arrow 21) the operation of the stage S_i (see block 22), so that the mobile agent a_i is executed in the places p_i^j of the stage S_i sequentially. As soon as one of the places p_i^j successfully executes the mobile agent a_i, this is recognized by the block 20 of the FTE (see arrow 23). This successful place is agreed upon among the set M_i of places p_i^j and is then called the primary place p_i^{prim}.

The block 20 of the FTE then confirms to all places p_i^j of the stage S_i that the primary place p_i^{prim} is committed and that all other places have to abort and/or undo any operation in connection with

the mobile agent a_i.

Except for the primary place p_i^{prim}, any operation in connection with the mobile agent a_i is then aborted and/or undone (see block 24 and block 25). As soon as this phase is finished, this is recognized by the FTE (see arrow 26).

The decision of the agreement problem of the current stage S_i is then present in the FTE (see block 27). This decision was already described above. The aforementioned primary place p_i^{prim} is identical with those places of figures 1a and 1b which have successfully executed the respective mobile agent a_i. In particular, with regard to figure 1b, the primary place p_i^{prim} of stage S_2 is the successful place p_2^1 and not the failing place p_2^0.

The block 27 of the FTE then moves the resulting mobile agent a_{i+1} together with the generated decision, in particular together with the set M_{i+1} of the places p_{i+1}^j of the next stage S_{i+1} to this next stage S_{i+1} (see arrow 28). This move of the resulting mobile agent a_{i+1} is performed as a reliable forward function.

For that purpose, each place p_i^j of stage S_i sends a clone of the resulting mobile agent a_{i+1} to all places p_{i+1}^j of the stage S_{i+1}. In order to reduce communication overhead, it is possible that only the primary place p_i^{prim} of the stage S_i sends the resulting mobile agent a_{i+1} to all places p_{i+1}^j of the stage S_{i+1} and that all other places of the stage S_i only verify whether the resulting mobile agent a_{i+1} has arrived at the places p_{i+1}^j of the stage S_{i+1}, e.g. by accessing the corresponding value in a repository of these places p_{i+1}^j.

As shown in figure 2, the block 20 of the FTE then starts to solve the agreement problem for this next stage S_{i+1}.

The described consensus method is implemented with a so-called agent-dependent architecture. As shown in figure 3, the FTE is integrated into the mobile agent a_i and travels with it to the sequential places p_i^j. Only one instance of the FTE exists per mobile agent a_i which is initialized by the user-defined agent 30 at the source of the mobile agent a.

The FTE is composed of a stage agreement component 31, a reliable forwarding component 32 and a recovery component 33. The stage agreement component 31 performs the consensus method, the reliable forwarding component 32 is responsible for reliably forwarding the resulting mobile agent a_{i+1} to the next stage, and the recovery component 33 handles any necessary recovery in case the mobile agent a_i fails or arrives too late at one of the places p_i^j.

The FTE provides a FTE-specific application programming interface 34 for the communication with the user-defined agent 30. The respective place p_i^j provides a repository 35 and further services 36. The repository 35 is a location where place-specific information may be stored temporarily. For example, the decision generated by the FTE may be stored in the repository 35, in particular the primary place p_i^{prim}. This information can then be kept until all other places of the respective stage S_i are aware of this decision. The information may then be discarded after a certain time.